

59

AN UNRESTRICTED TEXT-TO-SPEECH ALGORITHM  
FOR THE VOTRAX SYNTHESIZER

Stephen Bradly Stein

A Thesis

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements  
for the degree of Master of Computer Science at  
Concordia University  
Montreal, Quebec, Canada

March 1982

© Stephen Bradly Stein, 1982

## ABSTRACT

### AN UNRESTRICTED TEXT-TO-SPEECH ALGORITHM FOR THE VOTRAX SYNTHESIZER

Stephen Bradly Stein

A new text-to-speech algorithm for the Votrax synthesizer based on stress is described. It converts a written word into speech in two stages; the assignment of stress and the translation of the graphemes into phonemes.

Stress assignment is based on internal affixes and syllable count. The external affixes, which do not affect stress, are removed. Syllable count is determined from the number of vowels, diphthongs and silent "e"s detected. Following this, the graphemes are translated into their phonetic representation via letter-to-sound rules. A dictionary is used for words which do not obey these rules. This method gives 100% correct pronunciation to the most frequent 5,000 words in the Brown corpus.

An experiment was conducted to test the intelligibility

of the system against a human voice and to see if it would do better than a much simpler system; the "Type 'N Talk" synthesizer, incorporating its own algorithm without stress assignment, produced by the same manufacturer of our Votrax synthesizer. Two different sets of material were used, a paragraph from Time magazine and nine lists of ten phonetically-balanced sentences.

Our new algorithm scored much higher on the intelligibility test than the "Type 'N Talk" synthesizer. The maxima of words correctly comprehended were: 27% for the "Type 'N Talk", 66% for the new algorithm and 96% for human speech.

An analysis of variance indicated that the increase in intelligibility observed for all systems over time was significant to the 0.01% level. The results for the paragraph were much better indicating the value of contextual information.

#### ACKNOWLEDGEMENTS

I am deeply indebted to my thesis adviser, Prof. C. Y. Suen, for his expert guidance and advice throughout the course of this research.

The advice given dealing with the experiment in the latter part of this thesis by Prof. N. Segalowitz was greatly appreciated, as was his aid in producing the spectrographs used in this thesis. Special thanks go to Captain C. Y. Laporte of the Royal Military College of Saint-Jean for allowing us to conduct the experiment at the college and for his efforts in rounding up the subjects. Gratitude is expressed to all those cadets who took part in the experiment.

---

Thanks are also extended to Tammi Rossman and Gary Libben for their linguistic advice and moral support.

Sections of this thesis were proof read by my brother, Barry and H. Namer and this was greatly appreciated. Last, but not least, I wish to thank Arlene, my fiancée, for her patience and help in correcting the dictations.

## CONTENTS

ABSTRACT . . . . .	iii
ACKNOWLEDGMENTS . . . . .	v
I. INTRODUCTION . . . . .	1
Methods of Synthetic Speech Production . . . . .	3
Digital Speech . . . . .	4
The Human Vocal Tract . . . . .	6
The Votrax Synthesizer . . . . .	7
II. EXISTING SYSTEMS . . . . .	11
A System Based on Morphs . . . . .	17
III. THE ALGORITHM . . . . .	24
Syllabification . . . . .	25
Stress Rules . . . . .	29
Text-to-Speech Rules . . . . .	40
Lexicon and Prelexicon . . . . .	44
Intonation . . . . .	46
IV. PERFORMANCE . . . . .	50
The "Type 'N Talk" Synthesizer . . . . .	52
Design of the Performance Test . . . . .	53
Results . . . . .	56
Sources of Errors . . . . .	60
C. S. P. Versus "Type 'N Talk" . . . . .	63
V. CONCLUDING REMARKS . . . . .	69
Further Improvements and Research That Could be Done . . . . .	70
REFERENCES . . . . .	72
APPENDIX . . . . .	76

## LIST OF FIGURES

3.1	Example 1 (using the word "shameful") . . . . .	48
3.2	Example 2 (using the word "computers") . . . . .	49
4.1	Percentage of Correct Words for Each System in Graph Form . . . . .	59
4.2	Spectrograph of the Speech Produced by the C.S.P. System . . . . .	65
4.3	Spectrograph of the Speech Produced by the "Type 'N Talk" System . . . . .	66
4.4	Spectrograph of Human Speech . . . . .	67

## LIST OF TABLES

3.1	Definition of Long and Short Pronunciation . .	44
4.1	Distribution of Lists Among Groups . . . . .	54
4.2	Percentage of Correct Words for Each System .	56
4.3	Performance on the Passage . . . . .	58
4.4	Percentage of Subjects Correctly Identifying Each Word in the Sentence and the Word's Vowel Duration (in milliseconds) . . . . .	68

## CHAPTER I

### INTRODUCTION

Speech is one of the fundamental modes of human communication. Even in countries where a high rate of illiteracy exists, the prime mode of the inhabitants' communication is via speech.

In one study by Ochsman and Chapanis [14] sixty groups, each consisting of two persons, were given a problem to solve co-operatively. They were allowed to communicate only in the manner of communication assigned to them. Interestingly enough, those groups instructed to communicate only by handwriting or typewriting took more than twice as long to solve the problem as did the groups utilizing only vocal communication. Furthermore, most of the time consumed by the speech communicating groups was spent in active communication.

The researchers drew two general conclusions which I believe everyone knows intuitively. No matter as to whether a person is sending or receiving a message, speech communication allows other activities to be done

simultaneously. Deeper concentration on the communication task is required when using hard copy methods, thereby increasing the mean time to solve a problem.

It is therefore reasonable to conclude that adding the capability of speech to machines would greatly improve the man-machine interface.

In this thesis I examine one aspect of this problem, speech synthesis by machine. No doubt, synthetic speech has many applications as seen presently with the proliferation of talking toys, watches, calculators, car instrument panels and yes, even microwave ovens. Speech adds a novel touch to these devices, although seldom altering their functional capabilities.

Synthetic speech unfolds at least one salient possibility. It allows the blind as well as various other disabled individuals access to the written word via reading machines, without transformation or alteration of the text. Synthetic speech also shows promise as a teaching method for our children.

The research in this thesis has been to implement a synthetic speech system using a commercially available synthesizer that would have an unlimited vocabulary. Upon completion, it could be used as part of a reading machine

for the blind and other applications as cited.

### Methods of Synthetic Speech Production

There exists numerous techniques to produce computer speech varying both in quality and complexity of implementation.

One of the first and easiest ways is to record the speech utterances on a device similar to an audio tape recorder. The speech is taped on one track of the tape and start and stop marks are placed on another. When the computer wishes to verbalize a point it simply reads the tape at high speed until it finds the start mark of the sentence or phrase it wants. Once the phrase is found, it places the tape recorder in play mode and reads the other track for the stop mark. Once found, it stops the tape recorder.

Certainly this method is crude, but it can produce highly intelligible and natural sounding speech. The word "can" is used because this is only true if the entire phrase is recorded as a whole, as opposed to the machine stringing together words recorded separately. In the latter case, the speech will sound lifeless and dull due to the lack of control of the suprasegmental aspects of the speech, such as

4  
rhythm and intonation over the entire utterance.

Other potential problems of this method include the lack of electromechanical reliability and the degradation of speech quality as the magnetic media wears out.

A variation of this system was the IBM 7770 Drum System which used a magnetic drum rather than a tape.

### Digital Speech

Another relatively simple and much more reliable method involves digitizing the speech and storing it in solid state memory. This is accomplished by playing the speech, through an analog to digital converter with the output directed to a computer. When the computer wishes to speak, it simply invokes the reverse process, taking the digital representation of the speech from its memory and playing it back through a digital to analog converter with the output driving an ordinary audio speaker.

Such a technique produces high quality speech with a high degree of reliability. Unfortunately, due to a required bit rate of approximately 30 to 100k bits per second an immense memory is needed which proves practical only for the smallest of vocabularies. A speech wave being

composed of a few basic repeating waveforms is highly redundant. Linear predicative coding makes use of this fact and eliminates the superfluous and stores only the essential data in the form of linear predicative coefficients.

This method still produces high quality speech utilizing much less memory than the preceding method. The bit rate used can be as low as 1200 bits per second, but 2400 bits per second would be the norm.

All the above methods discussed so far are restricted to a finite vocabulary. They can only produce utterances that have been predetermined. Additionally, each new utterance added to the computer's repertoire would require additional memory space.

In certain applications such as toys, test equipment, flight simulators, weather reports, talking clocks, and the like, where the vocabulary is fixed and of a moderate length, these limitations are not important. It probably makes good sense to use linear predicative coding for these applications considering chips are available from Texas Instruments especially for this purpose. (One such chip is the TMC0281NL, a digital signal processor containing timing and decoding circuits, a 10 pole digital lattice filter and a digital to analog converter. Another chip is the controller chip TMC0271NL which does the mathematical

calculations required in linear predicative coding.)

One final word about the above methods is in order. If it is decided to make additions to the vocabulary the new recording must be done by the same individual who originally recorded the words. If that individual is not available, then the entire vocabulary must be redone!

There exists many applications where an unlimited speech vocabulary is needed, such as in reading machines for the blind, large and varied data bases, or for a truly flexible man-machine interface. (Of course, one would also need unlimited speech recognition.)

### The Human Vocal Tract

It would seem reasonable that one way to obtain unlimited speech would be to produce a synthesizer based on the human vocal tract.

Humans produce sounds by creating periodic and random acoustic excitation within the vocal tract. As air from the lungs is forced through, the vocal chords begin to vibrate causing periodic acoustic energy which is termed 'voiced' energy. Any sound which is produced with the vocal chords vibrating is termed voiced, as in the long "V" sound.

Conversely, any sound produced with the vocal chords open is referred to as 'unvoiced', such as the long "f" sound. This air then passes over the articulators (i.e. teeth, tongue, lips, etc) to produce random acoustic excitation or as it is more commonly called, frication.

One must realize that the entire vocal tract consists of many resonant cavities which act as filters. As the acoustic energy passes through peaks, commonly called formants will be formed at or near the resonant frequency of that part of the vocal tract. Of the series of formants formed, only the three lowest in frequency need be varied to produce intelligible speech.

For each new sound that one wishes to utter the articulators must be repositioned. As this repositioning occurs the frequency response of the vocal tract will change smoothly, rather than abruptly, because of the articulators' smooth movement from one state to another. This means that each sound produced is influenced by what occurred before it (dynamic articulation).

### The Votrax Synthesizer

From the above brief description of human sound production we can view it as nothing more than a series of

filters and acoustic energy sources.

The Votrax synthesizer uses this fact to become an electronic analog of the human vocal tract. It consists of two sound generator circuits. One produces voiced sounds and the other produces fricative sounds. These two outputs are joined and passed through a set of filters to simulate the vocal tract's resonance.

As it stands the parameters driving the above circuits would have to be updated every 5-25 milliseconds. However, the Votrax has some additional circuitry, the parametric control unit, that eliminates the need to update these parameters. This circuit controls and updates all the parameters needed to produce any of sixty-one phonemes. (Votrax literature, for models VS-6.0 and SC-01, says sixty-three, but two of them are pauses.)

Finally, there exists a dynamic articulation control unit between the parametric control unit and the remainder of the circuitry. It serves to modify the parametric control output to account for the movement of the articulators between phonemes.

The entire synthesizer is hardware controlled and all that is required to run it is a stream of phonetic codes represented in six bit words. Certain Votrax models allow

other parameters to be controlled. Model VS-6.0, the one which is used for this research, has four set inflection levels which change the phonemes' pitch and amplitude, requiring an additional two bits. Therefore, the entire command word has eight bits.

The bit rate needed to drive these synthesizers is only about 70 bits per second and it uses much less memory than any of the other methods.

One drawback of this synthesizer is that the sound quality is not nearly as good as the other methods, although it is intelligible. The lack of software control of the pitch, speech rate, and amplitude are serious shortcomings when trying to control the suprasegmental aspects of speech, mainly because it is nearly impossible!

Nevertheless, the Votrax synthesizer was chosen for this research, because it requires a low bit rate to operate and with suitable programming it can be made to speak unlimited English. Although it was nearly impossible to synthesize the prosodic elements of speech using this synthesizer, it would nonetheless allow for the production of a small unlimited speech synthesis system.

Generation of this set of rules is no easy task. It is well known that the orthographic representation of English

words does not always coincide with its correct pronunciation. One letter may map to many different sounds. It is this problem that this thesis addresses. (See chapter 3 for the algorithm.)

Although the software was developed on a large mainframe, CDC's Cyber 172, there is no reason why it cannot be made to operate on some of today's newer microcomputers. Although a Votrax VS-6.0 was used, Votrax produces a single-chip synthesizer, the SC-01, which is virtually identical to the VS-6.0, which can be used instead.

## CHAPTER II

### EXISTING SYSTEMS

There exists a number of speech by rule systems, some of them being simple, with others being far more complicated.

The methods employed by humans to produce speech from its orthographic representation are not fully understood. Furthermore, a system that employed all these things, if they were all known, would surely be of an incredibly large size. The particular problem that presents itself, is the selection of an algorithm of a suitable size to be run on a small computer, without overtaxing its processing capabilities. If this criterion cannot be met, then it is most likely the resultant system would be far too expensive for the average person to acquire.

With the availability of small inexpensive microprocessors it would be sufficient to have an algorithm that could run on one of these, and dedicate the microprocessor for this particular application.

The problem remains to find an algorithm that is sufficiently compact to run on one of these microprocessors, yet produce acceptable synthetic speech results using the Votrax synthesizer.

Elovitz et al [5], at the Naval Research Laboratory in Washington DC, wrote a very simple program to run the Votrax VS-6.0 synthesizer. The entire method is based on 329 letter to sound rules exclusively. In this approach, the text is scanned from left to right, and for each character scanned, the rules are sequentially searched until rules that are relevant to that character are found. Once a relevant rule is found the phonetic sound, in IPA, for that letter or letters is sent to a buffer. There is no provision for exceptions; a rule must be incorporated into the list of rules for each exception.

Lastly, the output of these rules is passed through another set of rules, similar to the letter to sound rules that translate the IPA symbols to the Votrax phonetic codes.

Such a system is very compact and easily implemented on practically any microcomputer. The drawback is that it is not all that accurate and suffers from a lack of intelligibility. One particular problem with the system is the lack of stress rules. Correct pronunciation of English words necessitates stressing the correct syllable of

polysyllabic words. Interestingly enough, the system on which Elovitz first based his approach on, Ainsworth's [1,2], does have such rules. However, Ainsworth used a parametric synthesizer which gave him control over the actual frequency each phoneme was given and its duration. Elovitz used the Votrax synthesizer and could not accomplish this in the same manner. In part, stress can be realized on the Votrax (see chapter 3), but perhaps, the method is not readily apparent.

In any case, Ainsworth's stress rules are very simplistic and far from complete. Stress is determined in the following manner. Each word is checked to a list of closed class words (articles, prepositions, conjunctions, etc), and if it is found in this list no stress is assigned. If not found, it is then checked to see if it contains a prefix, if so, its second syllable was stressed. Otherwise, the first syllable is stressed.

This is hardly an accurate method for polysyllabic words, but is acceptable for mono and many bisyllabic words. It should be noted that this method does not necessitate doing syllabification, but merely looking for a vowel.

Ainsworth also included pauses between breath boundaries. Breath boundaries were defined at: 1) punctuation marks; 2) preceding a conjunction; 3) between a

noun phrase and a verb phrase; 4) before a prepositional phrase; 5) before a noun phrase; and 6) after 50 characters have appeared in the input without meeting any of the other conditions. As one may have expected, Ainsworth's system, like Elovitz's, is not too intelligible.

Both of the above programs suffer from a lack of intonation control. Intonation is an essential part of the English language and in some cases, it alone determines the meaning of a sentence.

A recent paper by Witten [23] describes a system that provides for intonation and rhythm of the sentence. The rhythm is assigned by a complicated set of rules and a look-up table. Intonation must be marked by the person entering the text. That is, the text must be input phonetically; syllable, word and phrase, and all boundaries, as well as pitch (intonation), must be marked by hand (1, 1+, 1-, 2, etc). Clearly, this is not automatic in that it relies too heavily on the knowledge of the person typing the text. It is therefore useless in the application of automatic reading for the blind. In fact, for the program to be of any use, a program needs to be written that performs the text to phoneme conversion and assigns the correct intonation.

There is no doubt that if we are aiming for natural

sounding speech, we must account for all the factors involved in a native speaker's linguistic competence. This is an extraordinarily difficult task because the phonological level of language interacts with both the syntactic and semantic levels. The interaction of these three systems occurs in such a way that ignoring the higher level hierarchies would not produce anything approximating native-speaker English. This involves complicated sets of rules, mainly because no simple set of adequate rules for describing either English syntax or semantics exist. Consequentially, any system for speech synthesis designed in this way is going to be more complicated than the systems considered thus far.

Furthermore, any system attempting to approximate native-speaker English for one reason or the other, must address problems that the other systems have chosen to ignore. For example, there is the problem of compound words which can engender a medial silent "e": The word scarecrow has a silent "e", as does "therefore", but most systems will pronounce this silent "e" because by virtue of the compounding that has occurred, the "e" is no longer considered final and is consequently no longer considered silent.

It would seem that this deficiency could only be avoided by employing a look-up table of all possible

compounds that have medial silent "e"s. This defect also creates another one for systems which deal with stress, for it will have a significant effect on stress placement, because stress placement is based on syllable count. The medial silent "e" will be seen as a syllable where it is not, the subsequent incorrect syllable count may cause incorrect stress placement.

There is another unit of speech that researchers might look at that will possibly yield better results. This is the morpheme. English words tend to have an internal structure, and their constituent parts are called morphs, which include prefixes, derivational suffixes, and inflectional suffixes. They are not limited to just these, but can be free, that is, a base word such as "truck" and "person". Or they can be a bound; words which must be combined with another morpheme. All English words consist of morphs.

This is the case because native speakers are much more attentive to and conscious of morphemes than phonemes. Morphemes are more crucial in maintaining the semantic continuity of an utterance; native speakers attend to strings of phonemes only so far as they are necessary in constructing these more meaningful morphemic units. Consequently, any system of rules which attempts to synthesize speech by considering the morphemic structure of

lexical items in a sentence, has come one step closer to approximating native speaker competence. In addition to being a more accurate model of the psychological reality of speech perception, such a system based on morphemes can also offer a certain facility and accuracy in stress assignment. This in turn, increases the intelligibility of speech produced.

#### A System Based on Morphemes

Setting up a system that uses the morpheme is not a simple matter. This is due to the fact that unlike Elovitz's or Ainsworth's systems, a system based on morphemes must necessarily involve many more complicated sets of rules. For this reason, it is natural that most researchers have steered away from such a task because they want something that is easily implemented. However Jonathen Allen [3] of M.I.T. has endeavored to produce such a system. This is one of the most advanced and interesting systems for synthesizing speech that exists.

Allen's rules fall into two main types: those used for the morpheme system and those comprising the letter-to-sound system. If a word contains a morph which is not recognizable, then it is transferred from the morph system to the letter-to-sound system, where the word is sounded out

in the way that a native speaker would when he or she encounters a new word.

In the morpheme part of the system an attempt is made to break up the word into its morphs. This is no simple task as can be seen by examining the word "resting". It could be broken into "re-sting" or "rest-ing". Obviously "rest-ing" would be the correct choice (or at least, most probable), and Allen's system will choose this one, because the inflectional affix (as opposed to the derivational one) is preferred in this case, as his rules realize. His system will also correctly pronounce the word "wound" in the following sentences: "I have multiple wounds" and "I wound up the string". The correct choice is made because there is a set of rules which realizes that "wounds" has only an inflectional affix whereas "wound" has two underlying inflections, that is, it is really "wind" and "ed" giving the past tense "wound". However, one can observe that these kinds of rules are very complicated.

In the above example, problems still remain when "wound" is a noun.

Once the pronunciation of the constituent morphs are found, they must be combined to form the completed word. Unfortunately, a straight concatenation of the phonetic transcription of the words as produced by morph analysis or

letter to sound rules cannot be executed, because the morphs are pronounced according to their location in the word. Thus, the system incorporates a complete set of stress rules which are used to determine the accurate pronunciation of morphemes word-contextually by making use of facts such as that suffixes can have a strong effect on the roots to which they are attached, (e.g. "felon/felonious" and "electric/electricity"). This fine tuning of the phonemic strings through morphophonemic and lexical stress rules is a very important difference between this system and others. However, notice that these rules can only be implemented once the words are broken up into their morphs. Accordingly, they cannot be easily implemented in either Ainsworth's or Elovitz's programs.

This type of fine tuning is often done by humans. Consider the first time you try to pronounce a word that is being read which you have never seen before: you tend to sound it out by breaking it up into parts. The sounding-out is done either by recognizing the internal morphs or by sounding-out the morphs which you have never seen before, by means of letter-to-sound rules. Then, once you have sounded out the parts you quickly repeat the word, attempting to correct any sounds which originally were said incorrectly. That is, you allow for changes in sound that take place due to the modification of the affixes. This is most apparent in young children when they are learning to read.

There are also rules that add the prosodic element to the synthesized speech. Naturally, there are also rules which calculate the parameters for the parametric synthesizer being used.

Whilst there can be no doubt that Allen's system is certainly a good one, incorporating all known aspects of speech production, it is not all that practical to implement on a small microcomputer.

The morph dictionary alone contains over 12,000 entries! In addition, there are all the other rules which require to be taken into account.

Therefore, this system cannot be implemented on equipment that the average individual could afford to buy, unless there is a huge market for it.

Price becomes a prime consideration in most cases. In particular, for a blind person wishing to work with computers, speech output would certainly make life easier. The present alternative, a much slower procedure, is the use of expensive Braille terminals.

If a talking terminal is inexpensive then a company hiring a blind individual would not object to buying one. Conversely, a very expensive one diminishes the likelihood

of the company obtaining one.

So far, the assumption has been that to be of any use computer speech must sound exactly the same as a native speaker. This is not necessarily true. If the application is not one of a life and death situation, mistakes in the English pronunciation is tolerable. Humans possess very versatile data gathering mechanisms and will readily correct many of the errors that the machine may make, within limits. Naturally, if most words are mispronounced then the system is useless. Absolute monotone, of course, should be avoided for long passages, but it may be tolerated in certain instances.

The algorithm for devices incorporating unlimited speech output, such as a reading machine for the blind, must be compact, yet fairly accurate on the more common words. This is necessary to keep the overall price down and make this device readily accessible to the general public. It may require some training, but it should probably not take more than an hour or so to become reasonably proficient at understanding the generated synthetic speech.

In the next chapter I would like to discuss a system that attempts to meet the above criteria. Elovitz's system is too simplistic. Allen's system is much too complex for practical purposes. The new algorithm presented herein is

considerably easier to implement than that of Allen's, but much more complex than that of Elovitz's.

A special referral must be made to a study which has just come to hand (in the course of this writing) by Susan Hertz [9], whose aim it is to produce a speech system that is accurate and compact. It incorporates a three-level strategy.

First, the word goes through a set of text modification rules that modify the orthographic representation of the word. For example, the "e" that is dropped when adding "ing" to care is added back. Also a feature matrix is associated with each word containing information as to whether each letter is vocalic or a consonant. This modified text is then passed through the conversion rules which transform it into a string of phonemes and an associated feature matrix. Information as to whether the sound is frontal, velar, alveolar, etc., is stored in the feature matrix. The resultant phonetic string is passed through the feature modification rules that add and delete segments and modifications to the feature matrix are made. It should be noted that these rules also mark the stress.

Finally, the string of phonemes and the feature matrix is passed through a set of rules that drive a parametric speech synthesizer.

This is an interesting system, but does require the employ of a parametric synthesizer, otherwise, the information in the feature matrix cannot be used. Although the system may be adequate, it does not easily lend itself to running the Votrax synthesizer.

Calculation of the parameters to drive the parametric synthesizer may be too much for a microprocessor to handle. It is interesting to note, that the result of passing a word through her text modification rules is similar to the first part of the algorithm used in this thesis, even though her method is different.

## CHAPTER III

### THE ALGORITHM

Initial attempts to formulate a set of text-to-speech rules disclosed that transcription of the vowels to their phonetic representation would pose problems if done only on the basis of graphemes. It appeared that much of the vowels' variations in pronunciation actually depended more on their stress than on its immediate segmental environment. With this in mind, it is self-evident that if the word stress can be determined, then its transcription should become more accurate with a reduced set of vowel rules. This is further reinforced with the knowledge that most unstressed vowels reduce to schwa and are therefore uniform across words.

The ability to assign stress denotes two aspects. First, the number of syllables can be determined for each word; second, that the capacity to assign the stress to the correct syllable exists regardless of any affixes attached to the root word.

Once the stress is determined a set of vowel rules must

employ these data and another set of rules must handle all consonants and unstressed vowels.

These concepts are basic to the text-to-speech algorithm.

First a word is read in. A word being defined as a string of letters delimited by any non letter, except "'" in certain cases. The word is then checked to a prelexicon and if found, its pronunciation is obtained from the prelexicon. If not, the process continues.

Syllabification is then performed, followed by external word affixes being removed. The prelexicon and the lexicon are now consulted to see if the root word is contained. If so, the root word's pronunciation is obtained from the lexicon. If not, stress is assigned.

The root word is then translated to a stream of phonemes utilizing the text-to-speech rules for stressed and unstressed vowels and consonants.

Finally, any affixes that were removed are glued back to the word in their phonetic representation. The affixes are transcribed to their phonetic realization using the same set of rules as the root word.

Now let us examine the process in more detail.

### Syllabification

Analyzing a word into its component syllables is an easy task for the trivial case of monosyllabic words and perhaps, for bisyllabic words. Beyond this, the process becomes more difficult because of the inconsistency of the English language. However, since the system needs to know only the vowels that are in different syllables, for stress purposes, the process becomes much simpler. In which case, we need not be concerned about clustering the consonants with the correct vowels. With this in mind, it is sufficient to define operationally syllable number as the number of vowel clusters in a word.

Accordingly, the number of syllables in a word can be determined by breaking up the word into vowel and consonant clusters purely on the basis of their occurrence in the word string. For example, the word "bookmart" would be divided into: /b/oo/km/a/rt/. There are exactly two vowel clusters obtained. This is precisely the number of syllables in the word. Words such as "creation" and "compute" would fail using this method. "Creation" becomes /cr/ea/t/io/n with two vowel clusters, but has three syllables. "Compute" becomes /c/o/mp/u/t/e/ with three vowel clusters, but only

has two syllables. To syllabify these words correctly necessitates having some rules which can determine which vowel groups are and which are not diphthongs. Furthermore, silent "e"s must be recognized.

To this end, a set of rules has been added which attempts to do this. However, these rules will not be correct for a medial silent "e" as in the word shameful. (Note: "Shameful" will still be pronounced correctly because it will be reduced to the root word "shame". Silent "e" will then be found. See figure 3.1 at the end of this chapter.) Thus, "shameful" will be seen as having three syllables when in fact it has only two.

The actual algorithm used, is as follows:

- 1) Check for word final silent "e"

"e" is silent if 1) "e" is not preceded by a, e, i, o, u or l

2) "e" is preceded by al, el, il, ol, ul or yl.

- 2) Break the word up into vowel and consonant groups.

(A vowel is any of a, e, i, o, u, and y when y is preceded by a consonant. Note that qu and gu are considered as one consonant. All other letters are consonants.)

- 3) Break a vowel group into two if the following conditions hold. (The break is made immediately after the first vowel of the vowel group.

Note: "\$" stands for a, e, i, o, u or y;

"!" any letter;

"\*" any consonant;

"#" either a, e, i, o or u.)

- i) A followed by O.
- ii) Y followed by I.
- iii) U followed by A and not in persua, gua, jau or qua.  
 U followed by E and in tuent, ruen, lu or luen.  
 U followed by I and in uine, uit\$, uing, lui or rui.  
 U followed by O and not found in quo or uor.  
 U in oui.
- iv) O followed by E and found in oel, oem, oet or oey.  
 O followed by I and found in oic.  
 Ô followed by U followed by I.
- v) I followed by U followed by M.  
 I followed by A and not found in liant, tian,  
 tial, riage, sian, lian, cial, liam, gia, nia,  
 sia, cial, liar, cian, tia or lia.  
 I followed by O and not found in cious,  
 xious, cion, shio, llio, vior, sion, tion,  
 tiou, nio or gio.  
 I followed by E and not in tience, cient,  
 nience, tient, !\*cienc, nient, riend, dier,

riez, ries, tier, view, piec, iel, ief, mie,  
hie, gie, iev, fie, ie, ieu or yie.

vi). E followed by A and found in #\*ea, creat, react or  
#cean.

E followed by O and found in eous, eol, eom, eod,  
neo or reo.

E followed by U and found in eus, eum or #reu.

### Stress Rules

After determining the number of syllables in any word the next step is to utilize this information to determine which syllable to stress.

Chomsky and Halle's [4] main stress rules figured prominently in considering the approach to take in stress determination. It could not, however, be applied directly because it placed much weight on the grammatical class of the word. This is unknown to the algorithm. Instead, a set of rules were formulated based on certain considerations about stress set down by Axel Wijk [21] in his book "Rules of Pronunciation for the English Language". His ideas were refined and formalized into a uniform set of rules.

However, before these rules are applied, all external affixes are removed. An external affix is defined to be an

affix which does not affect the placement of stress on the word.

This makes the assignment of stress easier because these affixes need not be considered. It also has a further advantage. It actually corrects mistakes that would normally be made in the syllabification of some words. For example, "basement" is seen as having three syllables due to the silent medial "e" being counted as a syllable. However "ness" is an external suffix so it would be removed, resulting in "base". "Base" would correctly be seen as a one syllable word because its final "e" will be seen as silent. Pronunciation of "basement" would now be correct because the system views the word as "base" and "ment" and will not pronounce the silent "e". Not only does the removal of external affixes make stress assignment simpler and more accurate, but helps detect silent medial "e".

Unfortunately, a word such as "bumblebee" will have its silent medial "e" pronounced, because "bee" is not removed. However, this is more the exception than the rule.

The affixes to be dealt with are those which are internal, now that the external affixes have been removed. Internal affixes, unlike external ones, do indeed affect stress. These affixes determine the word's stress depending on the number of syllables in the word. These affixes are

searched for in a systematic way and the word stress assigned.

Of course, there are rules which act on words which do not contain these affixes. They look only at syllable number to determine stress. The affix removal rules are as follows: (Note: If not otherwise stated, in rules which cut off affixes it is assumed that the entire affix is cut off. If this is not the case, then the part which will not be cut off will be enclosed in brackets. The rules are applied linearly. "#" means a, e, i, o or u. "\*" means any consonant. "\$" means a, e, i, o, u or y. "!" means any letter.)

1) For all words beginning with:

under	para	side
some	micro	north
east	south	west
up	down	over
house	sky	out
air	tele	after
there	mid	head
where	hydro	

Treat the word as two different words, provided that at least two letters follow the prefix.

## 2) For all words ending in:

iably	ings	ably
thing	ing	able

Cut off the suffix, add "ed" and consider the word like this for stress. Re-calculate the number of syllables the word has.

(Note: If after cutting off the suffix there remains no syllables then ignore this rule.)

## 3) For all words whose syllable count is greater than 2 and ends in:

ful	ness	(*)ly
(e)ly		

Cut off the suffix, re-calculate the number of syllables and change final "ie" to "y".

This rule is applied twice in succession.

## 4) For all words ending in:

cred	vited	qu#*ed
ued	*uned	*eted
created	uated	*#ded
*eded	uaded	uided
anged	enged	unged
*ined	*ired	*ared
*ured	tored	bored
cored	*ated	iated

nited	cited	*oted
*uted	*iled	oled
uled	#med	rged
#ped	dged	#ged
*#ked	kled	pled
zled	#sed	nsed
psed	rsed	dled
#bed	bled	tled
fled	vled	aled
thed	zed	ved
ied	ced	*ided

Cut off the final "d" and re-calculate the number of syllables. Change final "ie" to "y".

- 5) For all words ending in \$ed, drop the final "ed" from the word, re-calculate the syllable count. If it is less than one, disregard this rule.

- 6) For all words ending in:

hes            sses            xes

take off the final "es". Subtract one from the syllable count.

- 7) For all words, except those which are greater than three letters long or ending in ss, ending in:

\*s            eas            las

ras	*es	ees
oes	ues	ies

Remove the final "s" and change final "ie" to "y". Re-calculate the number of syllables.

8) For all words ending in:

time	self	more
teen	teenth	town
#by	*ety	what
\$one	how	where
day	ism	ment
doy	*son	land
less	man	ton
thing	body	ship
out	way	fare
cast	hood	#over
room	wood	house
ground	ball	ever

Remove the suffix from the word and change final "ie" to "y". Re-calculate the number of syllables in the word. If the number of syllables is less than one, ignore this rule.

Of course, an assumption has been made about external and internal affixes. That is, they can only be one or the other. This unfortunately, is not always the case.

In the first stress rule, the suffix "able" is removed with the assumption that it is an external suffix. Naturally, in the word "table" it is not. This fortunately, presents no problem because the algorithm will not drop the affix, if it reduces a word to zero syllables.

Consider "admirable" and "desirable". Both are four syllable words and have the "able" suffix. The problem is that the "able" suffix is internal to the word "admirable" and external to "desirable". Undoubtedly, exception rules, can be generated and of course, some were used.

Some of these rules do not necessarily reflect the manner in which native speakers organize their words, but they seem to work. The idea is to be sure that more words fit the rule rather than the exception, otherwise the rule must not be retained.

It is unacceptable under any circumstance to have a rule which is applicable to only one case.

Actual realization of stress presents a particular problem. Ideally, the stressed syllables should exhibit some rising and following of pitch as mentioned by Hyman [12]. In practice, this is not at all feasible using the Votrax VS-6.0. Its four levels of inflection are distributed much too narrowly over the stressed vowel and

the pitch change is far too great. To be done properly, pitch must be able to be varied by software in very small increments. This hardware limitation means that the stress can only be realized by changing the vowel duration. Most, but not all vowels can have one of four durations that are software selectable.

Even though the stress realization is crude, it appreciably improves the speech produced.

The stress rules are as follows: (Note: "\$" stands for a,e,i,o,u or y; "!" any letter; "\*" any consonant; "#" either a,e,i or u. The stress rules are applied in a linear fashion until the first rule that applies is found and the process terminates.)

- 1) For all three syllable words ending in:

ster	ual	y
am	ise	yz
ize	ate	ite
ect	ute	ude
cle	ist	enue
inent	anent	quent
ident		

Stress the first syllable of the word.

- 2) For all words of two or more syllables ending in:

ect	oy	een
eur	eer	oon
ette	esce	aire
tine	zine	rine
dine	chine	ina
ona	ana	ita
ever	ota	

Stress first syllable of suffix.

- 3) For all words of three or more syllables ending in:

go	ma	to
do	no	scent
dent	ic	que
ia	ion	ogy
sive	iod	ior
ity	ional	ioner

Stress the syllable before the suffix.

- 4) For all words beginning with "trans", stress the the first syllable of the word.

- 5) For all words of four or more syllables ending in:

icacy	imacy	inacy
mony	sy	ary
tery	ory	ator
ize		

Stress the fourth syllable from the end of the word.

- 6) For all words of four or more syllables ending in "al": If one vowel or two or more consonants precede the suffix, then stress the syllable before the additional vowel or before the two consonants. If not, then stress two syllables before the "al".

- 7) For all words of two or more syllables ending in:

ence	ency	ous
#cent	ant	sis
on	an	ar
um	us	a

If one vowel or two or more consonants precede the suffix then stress the syllable before the one vowel or two or more consonants. If not, then stress two syllables before the suffix.

- 8) For all words of two or more syllables ending in:

*le	am	or
tist	ail	em
ext	ex	*le
ile	er	ance
ie	el	iemnt
en	fort	ward

y [but not for "ply"]

Stress the first syllable of the word.

- 9) For all words of two or more syllables  
having the following prefixes:

super	sus	intro
inter	mis	cro
suf	en	with
com	con	de
dis	dir	div
em	ef	ig
ex	imp	in
ob	oc	opp
per	pre	pro
pur	re	suc
sug	ir	sup
sur	un	a
be	forg	

Apply this rule twice, to check for multiple prefixes,  
and stress the syllable after the last prefix found.

- 10) For all words of three syllables ending in:

ent /le

Stress the first syllable of the word.

- 11) For all words of four or more syllables  
ending in:

ture ible

Stress the fourth syllable from the end.

- 12) For all words of four or more syllables ending in "er": If one vowel or two or more consonants precede "er" then stress the syllable before the one vowel or two consonants. If not, stress two syllables before the "er".
- 13) For all words of two syllables stress the first syllable.
- 14) For all words of three syllables: stress the second syllable if the second syllable is heavy, else stress the first syllable. (Heavy means one or more vowels followed by two or more consonants.)
- 16) For all words of four or more syllables: stress the third syllable from the end of the word.
- 17) For all words that contain at least one syllable stress the first syllable.

#### Text-to-Speech Rules

Once stress has been determined, only one final action remains to be taken, that is, translation of the text to a

stream of phonetic symbols, which is achieved by use of text-to-speech rules. The rules are organized by letter and operate on the grapheme translating it to a distinct phoneme.

An attempt has been made to make the number of rules associated with each grapheme correlate with the number of phonemically different environments each letter possesses. That is, an indication of the grapheme's phonological complexity can be obtained by the number of rules that is required to realize all its possible pronunciations.

The appendix contains an explanation of the grammar used to describe a rule and some examples of the 448 letter-to-sound rules. There are 424 entries in the lexicon. Let us work through a word. The word "cat" will be translated in the following way; the "a" in cat will be marked that it should be stressed. Next, the word will be scanned letter by letter from left to right. First, it will look at "c" and try to find a "c" rule. Looking linearly through the "c" rules it will apply each rule until it finds one that fits. In this case, the rule [c]=/k/ is used. It simply means that the letter "c" is pronounced "k" in this environment. The letters to the right of the equals sign, "k" in this case, represent the Votrax sounds as marked on its keyboard and are not IPA symbols. The square brackets delimit the letter(s) that will be transcribed by the rule.

Next a rule for the letter "a" must be found among the stressed "a" rules. [a](cl)≠/ae/ is the one used. It implies that if "a" is followed by one or more consonants and they are word final then "a", because it is the only letter enclosed in square brackets, is pronounced "ae". Clearly this rule applies for "at" in cat.

So far we have translated "ca" as "k ae". The "t" rule used is [t]=/t/. Thus cat would be realized as "k ae t".

A rule can be much more complicated than the ones illustrated. For example, a rule such as /g;#be/^[g]/i;e/~/ll;rl/ /(vl);n;m;l;r;c;s;#/ = /d j/ is certainly valid and in fact one of the rules. It means that "g" or word initial "be" cannot occur before "g". However "g" must be followed by an "i" or "e". They cannot be preceded by "ll" or "rl", but must be followed by either: one or more vowels; "n"; "m"; "l"; "r"; "c"; "s"; or be word final. If the above holds true then the "g" is pronounced as "d j" as in gesture. The reader should note that as there are "a" stressed rules, also there are "a" non-stressed rules. Two sets of rules exist for each vowel, a set when it is stressed and another when it is not stressed. Actually, it is a bit more complicated than that.

The rules that exist under "a" stressed rules would more correctly be referred to as exception rules. (This

applies to the remaining vowels also). In fact, there exists a category of rules called general stress rules that apply to all stressed vowels. In reality, the following sequence of events will take place when a stressed vowel is to be translated to its phonetic description. For purposes of convenience, let us use the vowel "a". First, it is checked to the "a" stressed rules, if no rule is found it is compared to the general stress rules. If yet no rule is found, then the vowel is given its short pronunciation. An explanation of the structure of the general stress rules will serve to comprehend what is meant by 'short pronunciation'.

One of the general stress rules is  $[(v1,1)]\# = 1$ . Simply, it means that if the stressed vowel at which we are looking is word final then it is given its long pronunciation. The definition of long and short pronunciation is contained in Table 3.1.

The correct vowel sound is simply chosen from the table. Long "e" would be pronounced "e" as opposed to "eh" for short "e". This method allows one rule to cover all vowels and allows each to have a distinct sound.

TABLE 3.1

## DEFINITION OF LONG AND SHORT PRONUNCIATION

Vowel	Long Pronunciation	Short Pronunciation
A	A1	AE
E	E	EH
I	AH1 AY	I
O	O	AH
U	U	UH
Y	AH1 AY	I

The ordering of the letter-to-sound rules is very important. If any rule is out of order it may never be realized. Notice, in the rules there exists two rules [wr]=/r/ and [w]=/w/. The first one says "wr" is pronounced as "r" and the second one says the "w" is pronounced as "w". Obviously if [w]=/w/ occurred first then [wr]=/r/ could never be realized. Thus, the rules depart from the specific to the general.

Lexicon and Prelexicon

There are a number of words which are exceptions to the rules. That is, a rule formulated to cover the word, would only be applicable to that word. Now, because the rules must be searched in a linear fashion it would be a waste of time to include exceptions in the rules. Because of the

nature of the rules, these exceptions would generally have to be the first entries in the particular letter rule, which implies that this rule will be scanned every time a letter is encountered which is the first letter of the exception word.

Since we know this rule can only be executed given a particular word it makes more sense to place it in a dictionary or lexicon. Then a binary search of words starting with the same letter as the one in question, could be done to see if the word is or is not an exception. This method is more speedy than searching linearly through the rules. For this reason a dictionary has been included:

The word "injury" is in the lexicon. Note the word "injuries" is not. This is so because the lexicon is searched after external affixes are removed and "injuries" would have become injury after removing the suffix "s". "Injury's" pronunciation would be accessed from the lexicon and the pronunciation of "s" would have been obtained from the letter-to-sound rules.

Generally speaking, it suffices then to place only the root word in the lexicon.

One particular problem exists when a suffix is both external and internal as in the case of "land". Islands

presents a problem. It will be broken into "is" and "land". Even if it is put in the lexicon it will never be found because it will have become "is". This means the lexicon must first be searched. However, so few words fall into this category, less than forty, it makes more sense to define a prelexicon so that there are less words for which to search.

#### Intonation

As mentioned before, the Votrax synthesizer being used has very limited intonation control. Little can be done to add the prosodic element to the outputted speech. The speech produced is very monotonous and it is difficult to determine where a sentence ends and the next one begins.

In a bid to alleviate this problem the following was added; any time a period, exclamation mark or comma is encountered, the pitch of the word preceding the punctuation mark is lowered from the stressed vowel onwards. Pitch is raised for a question mark. The punctuation marks mentioned are transcribed as a pause.

Naturally, the pause generated for a comma is of a shorter duration than for the other cases. These pauses are in addition to the pause generated between words, which are

of a very short duration.

This minor enhancement has dramatically improved the overall quality of the speech for long messages.


This ends the explanation of the new algorithm. Figures 3.1 and 3.2 show an example of passing a word through the described algorithm. Notice that figure 3.1 contains the example of the word "shameful" described earlier in the chapter.

In the next chapter the performance of the algorithm presented here will be given. However, it should be noted that it is meaningless to discuss the accuracy of the component parts because individually they do not add up to the final accuracy of the system. This is because each part takes into account the errors that the other parts may make.

## FIGURE 3.1

EXAMPLE 1 (using the word "shameful")

WORD% 'SHAMEFUL'



C SH  
V A  
C M  
V E  
C F  
V U  
C L

THERE IS/ARE 3 SYLLABLE(S)

PREFIX/SUFFIX RULE(S) NUMBER(S)-3  
1 SYLLABLE(S) IS/ARE CONSIDERED IN RULE  
AFTER STRESS RULE, WORD BECAME% SHAME  
FIRST SYLLABLE OF THE WORD IS STRESSED% A  
RULE 26

MAIN WORD

[SH]=/SH/

[(V1,1)](C1,1)/Y;AIL;IA;IO;IU;EA;EO;EU;IE;AL#;  
OT#;IVE;ILE;OR#;OLAT;E/=L

[M]=/M/

(C1)[E]/#;S#/=//

SUFFIX

%[FUL]#=/F UH3 L/

## FIGURE 3.2

EXAMPLE 2 (using the word "computers")

WORD% 'COMPUTERS' ?

C C  
 V O  
 C MP  
 V U  
 C T  
 V E  
 C RS

THERE IS/ARE 3 SYLLABLE(S)

PREFIX/SUFFIX RULE(S) NUMBER(S)-7  
 3 SYLLABLE(S) IS/ARE CONSIDERED IN RULE  
 AFTER STRESS RULE, WORD BECAME% COMPUTER  
 STRESS 1 AFTER THE PREFIX %U  
 RULE 19.

MAIN WORD

[C]=/K/

[/O(S1);O/]=/UH3/

[M]=/M/

[P]=/P/

(C1)/TH;ST;SH;CH;N;X;Y;S;Z;J;L;R/^[U](C1,1)(V1)=/Y U/

[T]=/T/

[/E(V1);E/]=/EH3/

[R]=/R/

SUFFIX

% (C1)/T'/^[S]#=/Z/

## CHAPTER IV

### PERFORMANCE

The first five thousand word entries in the Brown Corpus [13] will be correctly pronounced by the system presented in this thesis, within the hardware limitations of the Votrax VS-6.0 synthesizer.

This was ascertained by listening to the synthesized version of each word and by looking at the phonetic transcription that was being produced to drive the synthesizer.

A word was deemed correctly pronounced if its pronunciation was similar to that of most Canadians or Webster's Dictionary.

It would have been desirable to have been able to check the phonetic transcription of each word directly to a dictionary, thereby giving an undisputed measure of accuracy. This cannot be done, because there is not an exact correspondence between Votrax phonetic symbols and IPA symbols. In fact, there is an algorithm that affects the

sound of the phoneme depending on its phonetic environment, which does not always do as you expect, built into the synthesizer.

The problem of vowel duration still remains. The Votrax VS-6.0 generally has at least three, sometimes four, different vowel durations. Only one of them is most suitable for any given word. Information as to vowel duration is not generally given in dictionaries.

The sum of the accuracy of the component parts of the system is not equal to the final accuracy of the system due to the system's design, as mentioned earlier in Chapter 3. For this reason, it is meaningless to give accuracies for the component parts, especially considering that they were not designed as stand alone units. (The accuracy of syllabification is approximately 90%, stress assignment is about 85% and the text-to-speech rules is about 92% on the first 5,000 words of the Brown Corpus.)

The accuracy of any synthetic speech system on the first five thousand words in the Brown Corpus is not necessarily a good indication of how it will be received by the general public. With this in mind, an experiment was set up to get an indication of the performance of the system as judged by naive users.

### The "Type 'N Talk" Synthesizer

Recently, another synthesizer was added to Concordia University's speech lab, the "Type 'N Talk". This synthesizer is based on the SC-01 chip and its repertoire of phonemes is nearly identical to the Votrax VS-6.0. Both synthesizers are made by the same company. However, the "Type 'N Talk" contains its own built-in algorithm for text-to-speech conversion. It is a very small inexpensive unit and therefore contains a minimal text-to-speech algorithm. Its low price would certainly make it a contender for use in a reading machine for the blind. Since it is using an almost identical synthesizer to the one used in this research, it was decided to include synthesized speech from this unit in the performance tests. Granted it employs an algorithm smaller than the one described in this research and therefore should probably not do as well. However, this is not the main concern.

The idea in this thesis was to produce a small algorithm that could be implemented on a microcomputer. If however, the new algorithm does not fare much better in the performance test than the "Type 'N Talk", it would indicate that a fairly simple algorithm is all that is needed for reasonable speech intelligibility.

### Design of the Performance Test

The aim of the performance test, in the context of this thesis, was to obtain an indication of the intelligibility of the speech generated by the system described in this thesis. From now on the system will be referred to as C.S.P. (Concordia Speech Project).

Since further work is planned on C.S.P. an experiment that generated much more data than required for the purposes of this chapter was done. This would allow for more complicated analysis to be performed at a later date.

The experiment consisted of giving a dictation of nine lists of ten individual, unrelated sentences. The sentences have the property of not allowing prediction of the first or last part of the sentence based on the other part. Every single English phonetic sound was contained in each list in the frequency that it would normally occur. These lists of phonetically balanced sentences are often referred to as Harvard Sentences [24].

The nine lists of sentences were presented to three separate groups of subjects. Three lists were spoken by a human, three by the "Type 'N Talk", and three by C.S.P.

Table 4.1 shows the lists presented. The order of

presentation of the lists to the subjects was list one followed by list two and so on. From this table it can be seen that the order of presentation of the different systems is counter balanced and that each list is dictated once by all three systems.

TABLE 4.1  
DISTRIBUTION OF LISTS AMONG GROUPS

Group	List								
	1	2	3	4	5	6	7	8	9
1	C	T	H	C	T	H	C	T	H
2	T	H	C	T	H	C	T	H	C
3	H	C	T	H	C	T	H	C	T

NOTE: H=Human; T="Type 'N Talk"; C=C.S.P.

After the dictation of nine lists of sentences, the subjects listened to a section of a Time magazine article of two hundred and forty-five words. Group one listened to the paragraph given by C.S.P. and group two was given the version produced by the "Type 'N Talk". Group three listened to the human version.

They were then asked to write, in one or two sentences, what the paragraph was about. Following this, they were

expected to give a numerical rating of the presentation of the paragraph.

The actual question asked was phrased as follows: "At the end of the paragraph you are about to hear would you please write one or two sentences that would summarize what the paragraph was about. Then, give a numerical rating between one and one hundred, which reflects how you felt about your own understanding of the paragraph and its presentation."

The experiment was conducted at the Canadian Armed Forces' Royal Military College of Saint-Jean. One hundred and twelve cadets ranging in age from seventeen to twenty-one were randomly divided up into three equal groups.

Although the participants of the experiment were supposed to have been native English speakers, who had never heard any synthetic speech before, some cadets not meeting these criteria took part. Their papers were eliminated leaving ninety subjects; twenty-five in group one, thirty-seven in group two, and twenty-eight in group three.

Each group was presented the dictation and the paragraph from a pre-recorded tape, in one of three identical rooms, at the same volume and speech rate of approximately 160 words per minute.

The dictation test for each list took approximately two minutes. A thirty second break was given between tests, meaning that each voice would be heard again in about five minutes. The entire dictation test took twenty-four minutes to complete.

### Results

Table 4.2 and figure 4.1 summarize the results of the experiment. When viewing table 4.2 it is important to remember that it is based on naive subjects having only listened to twelve minutes of synthetic speech by the end of the complete dictation test. In light of this, the results are encouraging for C.S.P..

TABLE 4.2  
PERCENTAGE OF CORRECT WORDS FOR EACH SYSTEM

System	Trial			Mean
	1	2	3	
Human	93	92	96	94
C.S.P.	55	59	66	60
Type 'N' Talk	16	18	27	20

Trial one is composed of the percent correct words that each group obtained on exposure to the first list dictated

by one of the three systems. Therefore, the percentage obtained for the human speaker in trial one is the sum of the percentage of correct words obtained in list one from group three, list two from group two and list three from group one.

The sums are taken this way to cancel the effects that the order of presentation of the lists may generate.

The same holds true for trial two and three except that trial two deals with the second exposure to a particular system and trial three deals with the third exposure to one of the three particular systems.

Figure 4.1 shows that the general trend was to improve over time for each of the three voices. Improvement was more pronounced for the synthetic voices than for the human.

A two way analysis of variance with factor trial(1st, 2nd, 3rd) and system("Type 'N Talk", C. S. P., Human) with repeated measure of both factors was done. The main effect for the system is  $F(2, 178) = 42.31, .001$ . The main effect for the three trials is  $F(2, 178) = 95, .001$ , while the interaction between the system and the trials is  $F(4, 356) = 9.96, .001$ .

In the case of the human it is most likely that the improvement is due to adapting to the testing procedure and

learning to write at the correct speed.

Whilst this may be a factor in the synthetic speech case, it is also most probable that the cadets were adapting to the synthetic speech.

TABLE 4.3

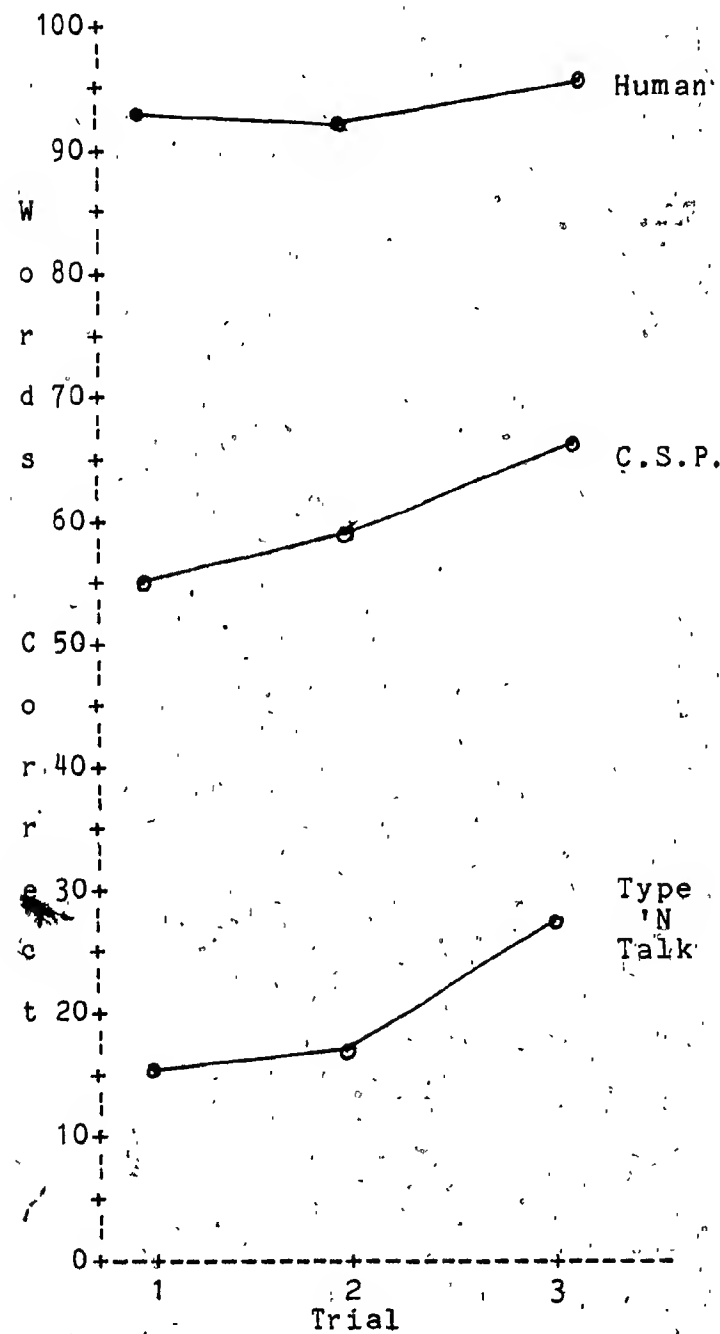
PERFORMANCE ON THE PASSAGE  
(scores given as percentages)

Scores	System		
	Human	C.S.P.	Type 'N Talk
Correct	100	97	71
Rating	91	64	32

Table 4.3 shows the results of the dictation of the passage. The passage was deemed correctly understood if the subject got the general idea of what it was about. The table also contains the average rating given to the system by the cadets, which roughly corresponds to their intelligibility scores.

These results would tend to reinforce the feeling that the synthetic speech system would do better on paragraph material than on single sentences containing little contextual information.

FIGURE 4.1  
PERCENTAGE OF CORRECT WORDS FOR EACH SYSTEM  
IN GRAPH FORM



Sources of Errors

Preliminary investigation into the causes of errors indicated that there are three main sources of errors; semantic, phonetic environment and pronunciation.

Semantic errors occurred in a great many of the lists given by C.S.P. and to a much lesser extent those given by the human. This was not that apparent in the lists done by the "Type 'N Talk" due to the limited number of words that the cadets actually comprehended and wrote down.

These errors manifest themselves as changes in words used that still produce a correct sentence. In the sentence, "The friendly gang left the drug store", "gang" was changed to "man" by a great many of the subjects listening to the C.S.P. version. One generally does not associate a gang as being friendly and would be more inclined to expect a man to be so.

"Hop" was changed to "jump" in the sentence "Hop over the fence and plunge in." by the same group of subjects. "Hop" and "jump" are highly associated words and the meaning of the sentence is not changed in this case.

Probably what is happening here, is that the subjects are understanding most of the sentence and attempting to

fill in any missing words according to the semantics of the sentence. In these sentences such a strategy will inevitably lead to error.

The phonetic environment also played a role in creating some transcription errors. If the sentence "The heart beat strongly and with firm strokes." is said quickly, it is inevitable that "beat" may be thought to be "beats". This occurs because the "s" in "strongly" that immediately follows "beat" may sound as though it is part of "beat" and "strongly". In the human case this is precisely an error that was made. It is unlikely that such an error would have been made if a time adverbial had been present.

The word "wood" in "Wood is best for making toys and blocks." was changed to "what" in many instances when it was presented by C.S.P.. This substitution is not all that odd. In isolation the "t" and "d" sounds would be pronounced differently. This is not the case in a sentence. When "t" is followed by a vowel, in this case the "i" from "is", the "t" is generally pronounced as a "d" because of intervocalic voicing. The word "butter" is a good example of this.

"Wood is best ..." is a strange way to formulate the beginning of a sentence, while "What is best ..." is not and is certainly more common. Given that the two sound very similar, it is not surprising that the substitution took

place rendering a different, but perfectly acceptable sentence.

Pronunciation obviously makes a difference. The word "tiny" was pronounced as "teeny" by the "Type 'N Talk" and not surprisingly not a single person got it correct. This is a case of straight incorrect pronunciation. (At least as far as Webster's dictionary is concerned.)

There exists a more subtle instance of this problem. "What" and "wood", from the above example, both contain lax vowels. Although there is a definite difference between the two vowels it is a subtle one, not that distinguishable in machine speech. In some dialects there is a distinction made between a "w" and a "wh" sound, but none is made by C.S.P.. It would also seem that none is made by "Type 'N Talk".

Preposition substitution, more specifically "a" for "the", took place in many instances in the human condition. In general, humans tend not to pay too much attention to prepositions unless they are important. Errors of this kind are therefore to be expected.

C.S.P. Versus "Type 'N Talk"

The large difference in words correct for C.S.P. and "Type 'N Talk" was not expected. It would be convenient to attribute all the difference between the two systems to the fact that C.S.P. uses a more sophisticated algorithm. Although this certainly makes a difference it should not account for all the difference.

Spectrographs taken of the sentence "The tiny girl took off her hat.", spoken by all three systems show some important differences between systems (figures 4.2, 4.3 and 4.4). The spectrographs were made from the tapes actually used in the performance test.

The formant transitions produced by the Votrax VS-6.0 synthesizer, used in the C.S.P., tend to be smoother than those of the "Type 'N Talk". Much more information is available from the speech produced by the Votrax VS-6.0 at the higher frequencies than that of the "Type 'N Talk". This is probably not attributable to the SC-01 synthesizer chip used, but rather to the amplifier used. It seems to have a sharp cut off point of about 3,000 Hertz. The amplifier is held suspect, because the SC-01 chip in the evaluation board supplied by Votrax sounds almost identical to the Votrax VS-6.0 synthesizer. Unfortunately, a spectrograph of the SC-01 chip in the evaluation board could

not be made at this time to verify this suspicion.

The spectrograph of the human is included for illustrative purposes. There can be little doubt that the human voice supplies much more information, although in this case not all of it is important. Table 4.4 shows the percentage of words that were correctly recognized by the subjects for each of the three systems. Little problem was encountered for the human or C.S.P. cases, but this was not the case for the "Type 'N Talk". The word "tiny" was not correctly recognized by any subject for the "Type 'N Talk" system, but it was probably due to the fact that it was pronounced "teeny".

Table 4.4 also shows the relative duration of each vowel for each word in the example sentence for the "Type 'N Talk", C.S.P. and human. The values in the table are only approximations, but the ratios between systems are correct.

The duration of the vowels is different for the three systems. The Vowels in "girl" and "took" were given shorter durations by C.S.P. than "Type 'N Talk". For all other words the reverse is true. This could be due to the lack of stress assignment of the "Type 'N Talk's" algorithm and the inclusion of stress assignment in C.S.P..

FIGURE 4.2  
SPECTROGRAPH OF THE SPEECH PRODUCED BY THE C.S.P. SYSTEM

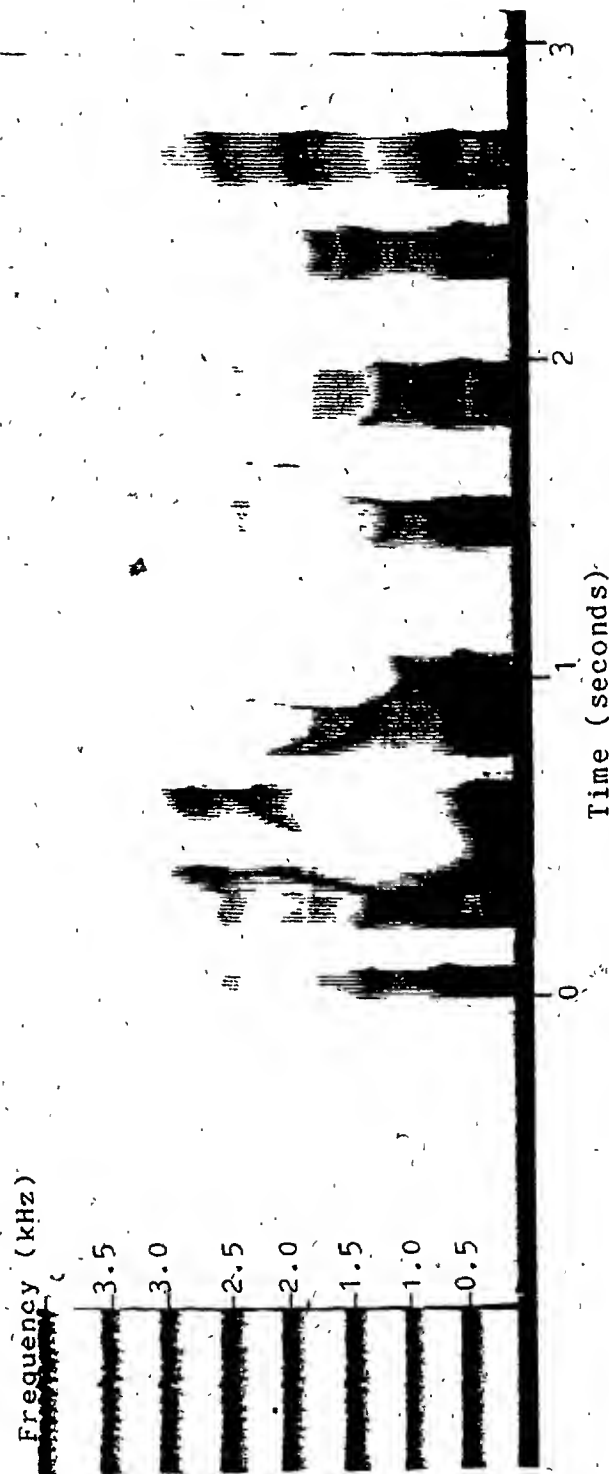


FIGURE 4.3

SPECTROGRAPH OF THE SPEECH PRODUCED BY THE "TYPE 'N TALK" SYSTEM

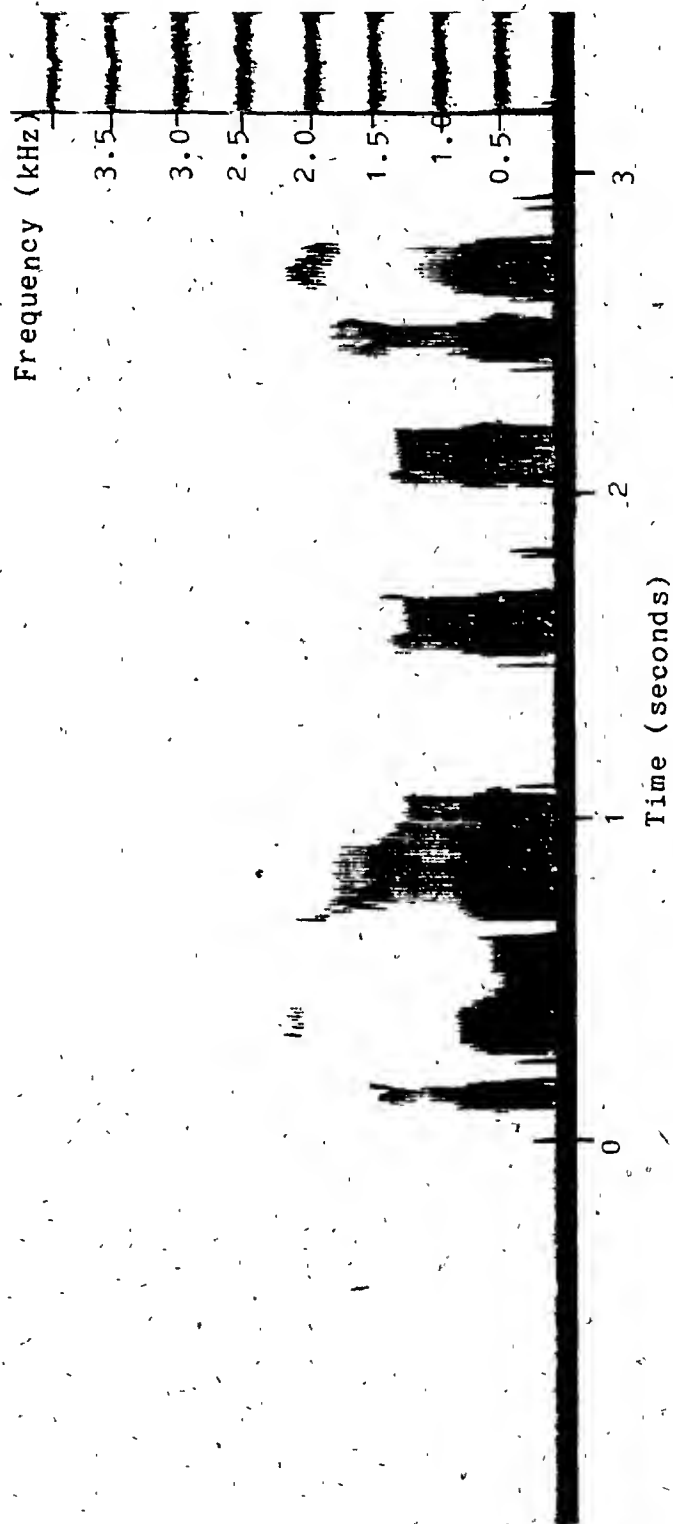
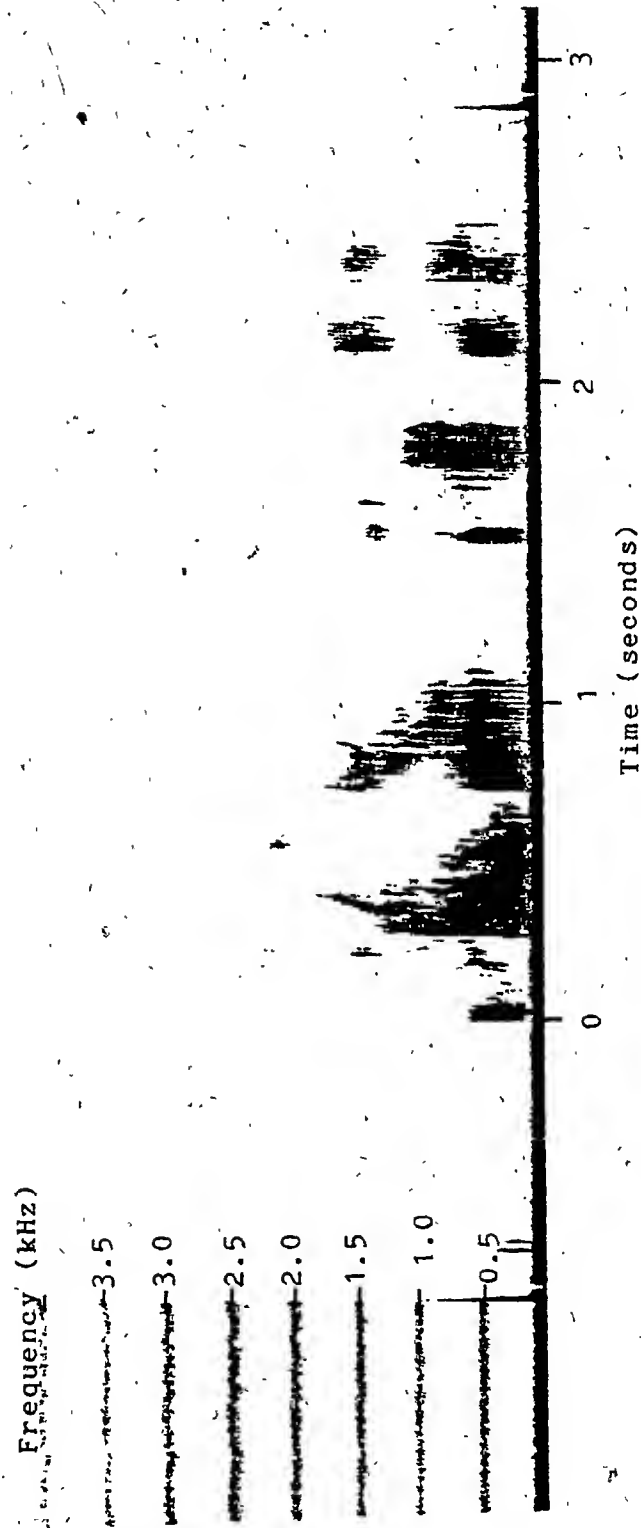


FIGURE 4.4  
SPECTROGRAPH OF HUMAN SPEECH



The human vowel duration is shorter than either synthetic voice systems. Although the vowel duration in the word "tiny" is the same for the human and the "Type 'N Talk" condition, it must be remembered that they said two different words.

TABLE 4.4

PERCENTAGE OF SUBJECTS CORRECTLY IDENTIFYING EACH WORD IN THE SENTENCE AND THE WORD'S VOWEL DURATION (in milliseconds).

System		Word							
		the	tiny	girl	took	off	her	hat	
C.S.P.	correct	93	93	100	100	100	100	100	
	duration	117	468	306	167	234	184	184	
Type 'N Talk	correct	64	0	76	52	60	68	36	
	duration	100	351	401	200	200	150	167	
Human	correct	100	92	100	100	100	100	100	
	duration	67	351	384	67	84	117	150	

## CHAPTER V

### CONCLUDING REMARKS

Synthetic speech has many diverse applications, many of which require an unlimited vocabulary. In the last few years various attempts have been made to formulate algorithms which would allow for an unrestricted vocabulary without actually attempting to store every word.

Algorithms ranging from very simple to extremely complex have been generated. The simple algorithms, or sets of rules, do not produce adequately intelligible speech. The most complex ones do produce fairly intelligible speech, such as MIT's MITalk system, but are too large to be of practical use at the present time.

In this thesis, a new algorithm has been presented which is fairly accurate, but still compact enough to run on very small computers. It extends on the idea of text-to-speech rules by adding the concept of stress realization. This also has the effect of alleviating some of the problems caused by silent medial "e".

Performance tests of the system indicate that it compares favourably to Susan Hertz's system [9]. Her system gets the first one thousand five hundred words in the Brown Corpus ninety six percent correct; C.P.S. gets the first five thousand words one hundred percent correct.

Although the percentage of correct words recognized for the Harvard sentences was not as high as one would have liked, it must be noted that there was definitely an improvement over time.

It is suspected that the score would improve appreciably over a few more hours. Persons who have been exposed to the system for a period of time tend to be able to understand it fairly well.

#### Further Improvements and Research That Could be Done

C.S.P could be markedly improved by taking into account the prosodic elements of speech. This would vastly increase the size of the algorithm and substantially increase its execution time. It would also necessitate the use of a parametric synthesizer.

Unfortunately, not enough is known about the way in which humans produce speech to formulate an algorithm

capable of generating perfect human like speech, with an unlimited vocabulary. Therefore, further research must be directed at obtaining a more complete picture of human speech production before machine speech will become almost indistinguishable from human speech. (Although this can be done with linear predictive coding, it does not allow for an unlimited vocabulary.)

In the meantime, the algorithm presented herein could be used in a low cost synthetic speech system. Although the speech produced is not completely natural, the important point is, it is intelligible. Furthermore, the algorithm is of a reasonable complexity such that it can be implemented on a fairly small computer.

## REFERENCES

1. W. A. Ainsworth, A system for converting English text into speech, IEEE Trans. Audio Electroacoust. AU-21, No. 3 (1973), 288-290
2. \_\_\_\_\_, Performance of a speech synthesis system, Int. J. Man-Machine Studies 6 (1974), 493-511
3. J. Allen, Performance of a speech synthesis system, Proc. IEEE 64, no. 4 (1976), 433-442
4. N. Chomsky and M. Halle, The sound pattern of English, New York, 1962
5. H. S. Elovitz, R. Johnson, A. McHugh and J. E. Shore, Letter-to-sound rules for automatic translation of English text to phonetics, IEEE Trans. Acoustics, Speech Signal Processing ASSP-24, No. 6 (1976), 446-458
6. K. Fons and T. Gargagliano, Articulate automata: an overview of voice synthesis, Byte 6, No. 2 (1981), 164-187

7. Foulds, Richard and Bronk, Spectrographic measurements of coarticulation in a commercial voice synthesizer, Proc. 4th Annual Conference on Rehabilitation Engineering (1981), 176-178
8. M. P. Haggard and I. G. Mattingly, A simple program for synthesizing British English, IEEE Trans. Audio Electroacoust. AU-16 (1968), 95-99
9. S. R. Hertz, SRS text-to-phoneme rules: a three level rule strategy, ICASSP 81 Proceedings 1 (1981), 102-105
10. D. R. Hill, using speech to communicate with machines, Infotech International Limited 1 (1979), 193-221
11. S. Hunnicutt, Phonological rules for a text-to-speech system, Technical Report, Research Laboratory of Electronics at MIT (1979)
12. L. Hyman, Phonology: theory and analysis, New York, 1975
13. H. Kucera and W. N. Francis, Computational analysis of present-day American English, Providence, 1967

14. R. B. Ochsmann and A. Chapanis, The effects of 10 communication modes on the behavior of teams during co-operative problem-solving, Int. J. Man-Machine Studies 6 (1974), 579-619
15. L. R. Rabiner, A model for synthesizing speech by rule, IEEE Trans. Audio Electroacoust. AU-17, No. 1, (1969), 7-13
16. C. Y. Suen, Recent advances in computer vision and computer aids for the visually-handicapped, Computers in Ophthalmology (1979), 259-266
17. C. Y. Suen, T. Rossman, S. Stein, M. G. Sobel, C. Charbonneau and L. Santerre, Computer speech synthesis at Concordia University, International Electrical, Electronics Conference and Exposition (1981), 176-177
18. N. Umeda, Linguistic rules for text-to-speech synthesis, Proc. IEEE 64, No. 4 (1976), 443-451
19. Votrax audio response system operators manual, Vocal Interface Division Federal Screw Works, Michigan, 1974

20. L. Walsh, Sound 1, Department of Linguistics, McGill University, Montreal, 1979
21. A. Wijk, Rules of pronunciation for the English language, London, 1966
22. Webster's third new international dictionary, Springfield, 1966
23. I. Witten and J. Abbess, A microcomputer-based speech synthesis-by-rule system, Int. J. Man-Machine Studies 111 (1979), 585-620
24. 1965 revised list of phonetically balanced sentences (Harvard sentences), IEEE Trans. Audio Electroacoust. AU-17 (1969), 238-246

## APPENDIX

### GRAMMER OF THE TEXT-TO-SPEECH RULES

In this appendix some examples of the text-to-speech rules are given. Chapter 3 contains an example of how they are used and this appendix is to be used in conjunction with it.

Each rule consists of two parts. The first part of the rule, to the left of the equal sign, indicates when the rule is to be used. While the right hand side of the rule indicates what phonemes are to be used if the rule holds.

Interpretation of a rule is done in the following manner. First, the characters within the square brackets are looked at to see if they are contained in a word. If so, then scanning continues in a left to right direction from the closing square bracket until the equal sign is found in the rule, or a character is found in the word, that does not match the one in the rule. If everything has matched so far, the scanning continues from the opening square bracket in a right to left direction until no match is found, or the end of the rule is found.

In some cases, instead of a letter, a symbol is used to represent a class of letters or a condition. The following list will explain all the symbols used:

% = Use this rule only for suffixes.

! = Any letter.

# = Word final or initial.

? = Any one vowel.

[ ] = Delimits the letters to be skipped if the rule holds.

= = Separates the matching part of the rule from the part which gives the phonetic sound.

/a1;a2;a3/ = Any one of the conditions a1, a2, or a3 must hold.

There is no limit on the number of conditions that may be used, except that a rule must fit on one line.

^ = It is used to compliment the above expression.

/B/^[A]/C//T/ means "A" not preceded by "B" or followed by "C", but must be followed by "T".

Remember the reversing scanning pattern when reading rules with "^" in them.

(ZI,M) or (ZI)

I = At least "I" Z's.

M = No more than "M" Z's.

Z can be: V = Any vowel.

C = Any consonant.

D = Any vowel provided it is  
not in the same vowel cluster

as the vowel immediately  
preceding this command.

S = Any vowel provided it is  
in the same vowel cluster  
as the vowel immediately  
preceding this command.

N = Any vowel which is not  
stressed.

Some example rules follow:

\*\*\*\*\*A RULES

%[ABLY]#=/UH3 B L E/

%[ABLE]#=/UH3 B UH3 L/

[AIL]=/A1 L/

[ATOR]#=/A1 T EH3 R/

[ATE]#=/A T/

[AY]=/A Y/

[AU]=/AW2/

[A]#=/UH1/

[/A(S1);A/]=/UH3/

\*\*\*\*\*B RULES

B[B]=//

[BODY]#=/B AH1 D E1/

%[BALL]#=/B AW L/

[BI]0=/B AH1 AY/

[BASIC]=/B A S EH3 K/

■ #[B]#=/B E/

[BEEN]#=/B I N/

[BROAD]=/B R AW D/

[BUIL]=/B I L/

[B]=/B/

\*\*\*\*\*GENERAL STRESSED RULES

[(V1,1)](C1,1)ER(V1)=S

[(V1,1)]~/ 'LL//X;LL;BLIC;TRIC;(C1,1)EDY#/=S

[I](C1,1)/IOU;IA;IO;IU;EA;EO;EU;IE/=S

[(V1,1)](C1,1)IT#=S

[(V1,1)]VE(C1)=S

[(V1,1)](C1,1)E/L;T/=S

[A]DE(C1)=S

[?](V1)=L

[(V1,1)]/#;'D#;GH;SURE;'LL/=L

[(V1,1)]SIS#=L

[(V1,1)](C1,1)/ATE#;IOUS/=L

[(V1)]~/ND/(C1)OUS=L

[(V1)](C1,1)US#=L

[/I;Y/](C1,1)R=L

[(V1,1)]~/NRY#/(C1,1)R(V1)(C0,1)#=L

[(V1,1)]CRE=L

[(V1,1)](C1,1)/Y;AIL;IA;IO;IU;EA;EO;EU;IE;AL#;

OT#;IVE;ILE;OR#;OLAT;E/=L

[(V1,1)](C1,1)LE#=L

[U](C1,1)(V1)=L